

# the Programmer

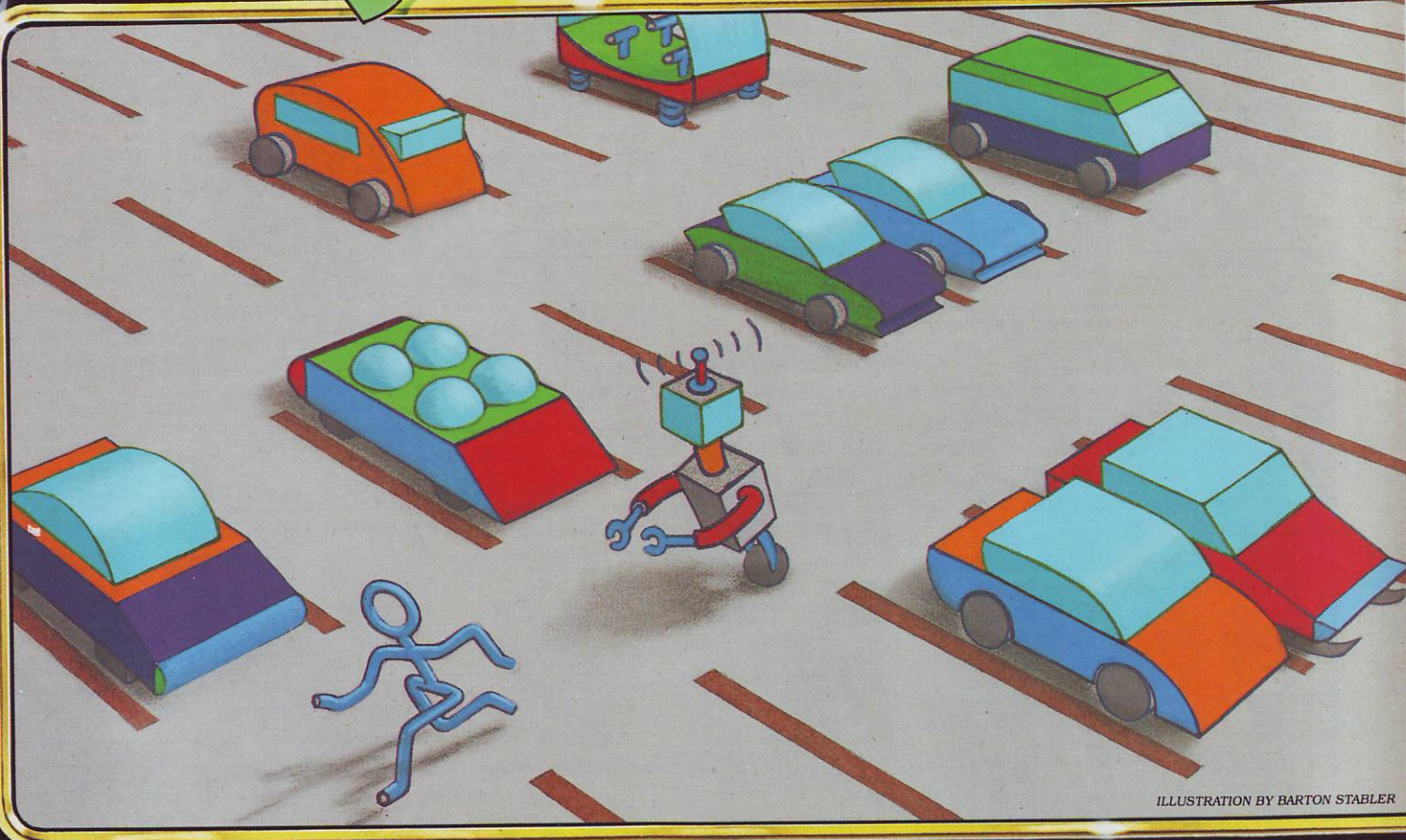


ILLUSTRATION BY BARTON STABLER

## A U G U S T

### TIPS TO THE TYPIST

Page 53

Helpful hints on how to type in programs, and what to do if a program doesn't work.

### BEGINNER PROGRAM

Page 54

Boogie down with *Jitterman*—and learn something about computer animation!

### ARCADE GAME

Page 57

Just when you thought it was safe to go back in the parking lot—it's *Renegade Robot II*!

### FEATURE PROGRAM

Page 61

Apple and Commodore 64 owners can get organized with *Home Information Manager*, an easy-to-use, data-base program.

**PROGRAMMING P.S.** Page 61 Corrections to previous months' programs and to a "Helpful Hint."

ILLUSTRATION BY JIM CHERRY III

Cherry



# TIPS TO THE TYPIST

Typing in FAMILY COMPUTING'S programs is a great way to become familiar with your computer and get some free software "to boot." But it's frustrating to type in a long program only to find it doesn't work as it should. When this happens, simple typing errors are most often the cause. So to help you gain greatest value from the time you spend computing with us, we've put together some tips on how to avoid typing errors—and what to do if a program doesn't run right. Read them carefully and you'll be up and running in no time!

## SOME GENERAL RULES

**1. Do** set up your computer in a well-lighted, comfortable location, and prop the magazine up so that you don't have to strain to read the printed listing.

**2. Do** read instructions and program headings carefully. Make sure your computer has enough memory, the right version of BASIC, and the appropriate peripherals (joysticks, printers, disk drives, etc.) for a program.

**3. Don't** let fatigue and boredom contribute to inaccuracy. If you're new to programming, try typing in shorter programs first. Type in a longer program in easy stages, **SAVING** each installment as you go.

**4.** Until you are fairly familiar with BASIC, **do** assume that every word, number, letter, space, and punctuation mark in a program listing must be copied accurately if the program is to function as intended.

**5. Do** watch out for potential trouble spots. About 90 percent of all typing errors occur in **DATA** statements: long lines filled with numbers or incomprehensible secret codes. If possible, have someone else read **DATA** to you as you type, and help you proofread it if you have trouble later on. Proofreading from a printout is best.

**6. Do** be aware that program listings printed in FAMILY COMPUTING sometimes differ from what you will see on your computer's screen or in printouts you produce at home. Our program listings are printed 54 characters wide. Thus, a single BASIC program "line" (sometimes called a "logical line") may appear as several lines in our listing. If you are typing along and reach the right margin of the printed listing, don't press **RETURN** or **ENTER** until you've checked to see if the program "line" you're typing really ends there. The way to tell is to check if the line following begins with a multiple of 10 that follows in sequence from the previous logical line. **REM** statements are the exception and typing them in is optional.

Several computers (ADAM, Apple, Atari, and TI) format BASIC programs according to unique rules of their own. Don't let this throw you—just type in the listing exactly as printed in the the magazine and your computer is guaranteed to accept it, even though it may end up looking a little different on your screen.

**7.** One foolproof way to correct an error in a BASIC program line is to type the line in again from the beginning, and press **RETURN** or **ENTER** to set it in place of the old one in your computer's memory.

## WHICH PROGRAMS WILL RUN ON MY COMPUTER?

Unless a program heading indicates otherwise, programs will run on any version of the computer specified, with the following exceptions:

- Apple programs run under AppleSoft (not Integer) BASIC on the Apple II (with language card), II plus, IIe, and IIc. The Macintosh is not supported as of this writing.
- Our Atari programs may in some cases be incompatible with the Atari 1200XL.
- IBM PC owners may occasionally require a Color Graphics Adapter to run our graphics programs.
- TI programs not marked "w/TI Extended BASIC" should be run under standard (console) TI BASIC.
- TRS-80 Model III programs will run on the Model 4 in Model III mode.

## DEBUGGING HINTS

Sometimes even the most careful typist makes a mistake. Don't expect your program to run right off the bat. If you have problems, remain patient and follow these general instructions for a probable quick fix.

**LIST** the program in screen-size chunks (check your manual for instructions on how to **LIST** parts of a program). Even better, if you have a printer, get a printout. Compare what you've typed in—letter by letter—to the published program. Make sure that you haven't typed the numeral 0 (which is slashed in our listings) for the letter O (which isn't), swapped a small letter "l" for the numeral one, dropped or mixed up some punctuation, switched uppercase text for lowercase or vice versa (particularly in **DATA** statements or within quotes), or miscounted the characters (and/or spaces) between a pair of quotes. Get someone to help you if possible.

Check your **DATA** statements—then check them again. Mistakes in **DATA** statements are the single most common cause of program failures. Bad **DATA** can cause a program to malfunction at any point, which can be misleading.

## WHAT TO DO WHEN YOU'VE DONE IT ALL

We're proud of our programs, and we want you to enjoy them as much as we enjoy writing them. If you just can't figure out what's wrong with a program, we'd like to help. But we can't if you don't provide us with important information. When you write us (no telephone calls, please), indicate:

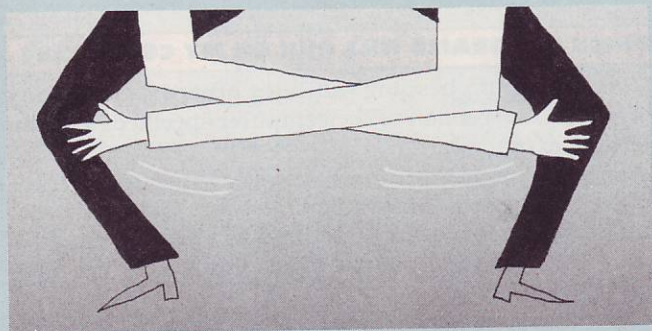
- Which program you're having trouble with.
- Which type of computer you own, the type of BASIC you are using, how much RAM your computer has, and what DOS and peripherals you're using, if any.
- What error messages your computer has given you.
- Your name, address, and telephone number.

If possible, please enclose a printout. Address all correspondence to Programming P.S., FAMILY COMPUTING, 730 Broadway, New York, NY 10003.



# JITTERMAN

BY JOEY LATIMER



The other day, while watching a "Gumby" rerun, I became fascinated with the dexterity used to animate that little clay figure. With the proper manipulations, Gumby appears to run, jump, crawl, bend, stretch, and dance. It's frightening how real he can be.

Computers weren't used to create Gumby, but they are used quite a bit today in animation of all kinds. This month's Beginner Program is a simple but semiserious attempt at computer animation.

Animation works by flashing a series of still images in front of your eyes fast enough so that differences between them are interpreted as movement. In conventional animation, these still images are called "cels", short for celluloid—the transparent material animators draw on.

In *Jitterman*, three groups of PRINT statements define three views of a dancing figure. These three "cels" are PRINTED one over another, creating the appearance of motion. As the program goes on, the cels are PRINTED further and further to the right—making the dancing man jitterbug across the screen. When *Jitterman* reaches the right side of the screen, he's blanked out, only to reappear at the left side.

*Jitterman* was designed to dance to almost any type of song. If you want, you can change the way *Jitterman* looks, or design your own animation "cels" by changing the characters

between the quotes in each block of PRINT statements. Nine PRINT statements make up each cel, and there are nine characters, including spaces, between the quotes in each PRINT statement.

*Jitterman's* speed is kept under control by letting a little time elapse between PRINTING each of his cels. The time-wasting is accomplished in each case by a pair of statements that look something like this:

```
FOR D=1 TO 50
```

```
NEXT D
```

You may recognize the familiar FOR/NEXT loop that BASIC uses for counting and doing things over and over. This kind of FOR/NEXT loop (termed "empty" because it contains no statements between the FOR expression and the NEXT expression) serves merely as a delay, counting from 1 to 50 before allowing the program to continue. "Delay loops" like this are often used to control timing in a program. You can speed *Jitterman* up or slow him down by changing the second number (e.g. the 50) in the FOR statement of each of the delay loops to a lower or a higher value.

If you experiment with this program and come up with something you're excited about, please drop us a line. If we like it, we might mention it in a future issue. Send a printout (no tapes or disks, please) and explanation to: Beginner Programs c/o FAMILY COMPUTING 730 Broadway New York, NY 10003

## Apple/Jitterman

```
10 HOME
20 VTAB 10
30 PRINT "-----"
40 FOR X = 1 TO 30
50 VTAB 1
100 PRINT TAB(X);" 0  "
110 PRINT TAB(X);" ### "
120 PRINT TAB(X);" # # # "
130 PRINT TAB(X);" # # # "
140 PRINT TAB(X);" #  "
150 PRINT TAB(X);" # # "
160 PRINT TAB(X);" # # "
170 PRINT TAB(X);" #  "
180 PRINT TAB(X);" V  V "
190 FOR D = 1 TO 16
200 NEXT D
210 VTAB 1
220 PRINT TAB(X);" <  "
230 PRINT TAB(X);" ##### "
240 PRINT TAB(X);" # # # "
250 PRINT TAB(X);" # # # "
260 PRINT TAB(X);" #  "
270 PRINT TAB(X);" # # "
280 PRINT TAB(X);" #  "
290 PRINT TAB(X);" # # "
300 PRINT TAB(X);" V  V "
310 FOR D = 1 TO 13
320 NEXT D
330 VTAB 1
340 PRINT TAB(X);" >  "
350 PRINT TAB(X);" ##### "
360 PRINT TAB(X);" # # # "
370 PRINT TAB(X);" # # # "
380 PRINT TAB(X);" #  "
390 PRINT TAB(X);" #  "
400 PRINT TAB(X);" #  "
410 PRINT TAB(X);" #  "
420 PRINT TAB(X);" V V "
430 FOR D = 1 TO 13
440 NEXT D
450 NEXT X
460 VTAB 1
470 FOR X = 1 TO 9
480 PRINT TAB(30);"  "
490 NEXT X
500 GOTO 40
```

## Atari/Jitterman

```
9 REM --"POKE 752,1" TURNS OFF THE CURSOR--
10 POKE 752,1
20 PRINT CHR$(125)
30 POSITION 0,10
40 PRINT "-----"
50 FOR X=0 TO 29
100 POSITION X,1:PRINT " 0  "
110 POSITION X,2:PRINT " ### "
120 POSITION X,3:PRINT " # # # "
130 POSITION X,4:PRINT " # # # "
140 POSITION X,5:PRINT " #  "
150 POSITION X,6:PRINT " # # "
160 POSITION X,7:PRINT " # # "
170 POSITION X,8:PRINT " #  "
180 POSITION X,9:PRINT " V  V "
190 FOR T=1 TO 3
200 NEXT T
210 POSITION X,1:PRINT " <  "
220 POSITION X,2:PRINT " ##### "
230 POSITION X,3:PRINT " # # # "
240 POSITION X,4:PRINT " # # # "
250 POSITION X,5:PRINT " #  "
260 POSITION X,6:PRINT " # # "
```



```

270 POSITION X,7:PRINT " # "
280 POSITION X,8:PRINT " # # "
290 POSITION X,9:PRINT " V V "
300 FOR T=1 TO 2
310 NEXT T
320 POSITION X,1:PRINT " > "
330 POSITION X,2:PRINT " ##### "
340 POSITION X,3:PRINT " # # # "
350 POSITION X,4:PRINT " # # # "
360 POSITION X,5:PRINT " # "
370 POSITION X,6:PRINT " # "
380 POSITION X,7:PRINT " # "
390 POSITION X,8:PRINT " # "
400 POSITION X,9:PRINT " V V "
410 FOR T=1 TO 2
420 NEXT T
430 NEXT X
440 FOR X=1 TO 9
450 POSITION 29,X
460 PRINT " "
470 NEXT X
480 GOTO 50

```

### Commodore 64/Jitterman

```

9 REM --SET SCREEN AND BORDER COLORS--
10 POKE 53280,0
20 POKE 53281,0
30 PRINT CHR$(147)
40 FOR X=1 TO 30
50 PRINT CHR$(19);
100 PRINT TAB(X);" 0 "
110 PRINT TAB(X);" ### "
120 PRINT TAB(X);" # # # "
130 PRINT TAB(X);" # # # "
140 PRINT TAB(X);" # "
150 PRINT TAB(X);" # # "
160 PRINT TAB(X);" # # "
170 PRINT TAB(X);" # # "
180 PRINT TAB(X);" V V "
190 PRINT "-----"
200 FOR D=1 TO 16
210 NEXT D
220 PRINT CHR$(19);
230 PRINT TAB(X);" < "
240 PRINT TAB(X);" ##### "
250 PRINT TAB(X);" # # # "
260 PRINT TAB(X);" # # # "
270 PRINT TAB(X);" # "
280 PRINT TAB(X);" # # "
290 PRINT TAB(X);" # "
300 PRINT TAB(X);" # # "
310 PRINT TAB(X);" V V "
320 FOR D=1 TO 13
330 NEXT D
340 PRINT CHR$(19);
350 PRINT TAB(X);" > "
360 PRINT TAB(X);" ##### "
370 PRINT TAB(X);" # # # "
380 PRINT TAB(X);" # # # "
390 PRINT TAB(X);" # "
400 PRINT TAB(X);" # "
410 PRINT TAB(X);" # "
420 PRINT TAB(X);" # "
430 PRINT TAB(X);" V V "
440 FOR D=1 TO 13
450 NEXT D
460 NEXT X
470 PRINT CHR$(19);
480 FOR X=1 TO 9
490 PRINT TAB(30);" "
500 NEXT X
510 GOTO 40

```

### IBM PCs/Jitterman

```

10 CLS
20 KEY OFF
30 WIDTH 40
40 LOCATE 10,1,0
50 PRINT "-----"
60 FOR X=1 TO 30
100 LOCATE 1,X:PRINT " 0 "
110 LOCATE 2,X:PRINT " ### "
120 LOCATE 3,X:PRINT " # # # "
130 LOCATE 4,X:PRINT " # # # "
140 LOCATE 5,X:PRINT " # "
150 LOCATE 6,X:PRINT " # # "
160 LOCATE 7,X:PRINT " # # "
170 LOCATE 8,X:PRINT " # # "
180 LOCATE 9,X:PRINT " V V "
190 FOR D=1 TO 3
200 NEXT D
210 LOCATE 1,X:PRINT " < "
220 LOCATE 2,X:PRINT " ##### "
230 LOCATE 3,X:PRINT " # # # "
240 LOCATE 4,X:PRINT " # # # "
250 LOCATE 5,X:PRINT " # "
260 LOCATE 6,X:PRINT " # # "
270 LOCATE 7,X:PRINT " # "
280 LOCATE 8,X:PRINT " # # "
290 LOCATE 9,X:PRINT " V V "
300 FOR D=1 TO 2
310 NEXT D
320 LOCATE 1,1,0
330 LOCATE 1,X:PRINT " > "
340 LOCATE 2,X:PRINT " ##### "
350 LOCATE 3,X:PRINT " # # # "
360 LOCATE 4,X:PRINT " # # # "
370 LOCATE 5,X:PRINT " # "
380 LOCATE 6,X:PRINT " # "
390 LOCATE 7,X:PRINT " # "
400 LOCATE 8,X:PRINT " # "
410 LOCATE 9,X:PRINT " V V "
420 FOR D=1 TO 2
430 NEXT D
440 NEXT X
450 FOR X=1 TO 9
460 LOCATE X,30:PRINT " "
470 NEXT X
480 GOTO 60

```

### TRS-80 Color Computer/Jitterman

```

10 CLS
20 PRINT @320,"-----"
30 FOR X=1 TO 21
40 PRINT @0
100 PRINT TAB(X);" 0 "
110 PRINT TAB(X);" ### "
120 PRINT TAB(X);" # # # "
130 PRINT TAB(X);" # # # "
140 PRINT TAB(X);" # "
150 PRINT TAB(X);" # # "
160 PRINT TAB(X);" # # "
170 PRINT TAB(X);" # # "
180 PRINT TAB(X);" V V "
190 FOR D=1 TO 13
200 NEXT D
210 PRINT @0
220 PRINT TAB(X);" < "
230 PRINT TAB(X);" ##### "
240 PRINT TAB(X);" # # # "
250 PRINT TAB(X);" # # # "
260 PRINT TAB(X);" # "
270 PRINT TAB(X);" # # "
280 PRINT TAB(X);" # "

```



## BEGINNER PROGRAM

```

290 PRINT TAB(X);"  # #  "
300 PRINT TAB(X);"  V  V  "
310 FOR D=1 TO 7
320 NEXT D
330 PRINT @0
340 PRINT TAB(X);"  >  "
350 PRINT TAB(X);"  ##### "
360 PRINT TAB(X);"  # # # "
370 PRINT TAB(X);"  # # #"
380 PRINT TAB(X);"  #  "
390 PRINT TAB(X);"  #  "
400 PRINT TAB(X);"  #  "
410 PRINT TAB(X);"  #  "
420 PRINT TAB(X);"  V V  "
430 FOR D=1 TO 7
440 NEXT D
450 NEXT X
460 PRINT @0
470 FOR X=1 TO 9
480 PRINT TAB(21);"  "
490 NEXT X
500 GOTO 30

```

### TRS-80 Model III/Jitterman

```

10 CLS
20 PRINT @640;"-----"
-----"
30 FOR X=1 TO 52
40 PRINT @0,"";
100 PRINT TAB(X);"  0  "
110 PRINT TAB(X);"  ### "
120 PRINT TAB(X);"  # # # "
130 PRINT TAB(X);"  # # # "
140 PRINT TAB(X);"  #  "
150 PRINT TAB(X);"  # #  "
160 PRINT TAB(X);"  # #  "
170 PRINT TAB(X);"  #  "
180 PRINT TAB(X);"  V  V  "
190 FOR D=1 TO 5
200 NEXT D
210 PRINT @0,"";
220 PRINT TAB(X);"  <  "
230 PRINT TAB(X);"  ##### "
240 PRINT TAB(X);"  # # # "
250 PRINT TAB(X);"  # # # "
260 PRINT TAB(X);"  #  "
270 PRINT TAB(X);"  # #  "
280 PRINT TAB(X);"  #  "
290 PRINT TAB(X);"  # #  "
300 PRINT TAB(X);"  V V  "
310 FOR D=1 TO 2
320 NEXT D
330 PRINT @0,"";
340 PRINT TAB(X);"  >  "
350 PRINT TAB(X);"  ##### "
360 PRINT TAB(X);"  # # # "
370 PRINT TAB(X);"  # # #"
380 PRINT TAB(X);"  #  "
390 PRINT TAB(X);"  #  "
400 PRINT TAB(X);"  #  "
410 PRINT TAB(X);"  #  "
420 PRINT TAB(X);"  V V  "
430 FOR D=1 TO 2
440 NEXT D
450 NEXT X
460 PRINT @0,"";
470 FOR X=1 TO 9
480 PRINT TAB(53);"  "
490 NEXT X
500 GOTO 30

```

### VIC-20/Jitterman

```

9 REM --SET SCREEN AND BORDER COLORS--
10 POKE 36879,8
20 PRINT CHR$(5)
30 PRINT CHR$(147)
40 FOR X=1 TO 12
50 PRINT CHR$(19);
100 PRINT TAB(X);"  0  "
110 PRINT TAB(X);"  ### "
120 PRINT TAB(X);"  # # # "
130 PRINT TAB(X);"  # # # "
140 PRINT TAB(X);"  #  "
150 PRINT TAB(X);"  # #  "
160 PRINT TAB(X);"  # #  "
170 PRINT TAB(X);"  #  "
180 PRINT TAB(X);"  V  V  "
190 PRINT "-----"
200 FOR D=1 TO 35
210 NEXT D
220 PRINT CHR$(19);
230 PRINT TAB(X);"  <  "
240 PRINT TAB(X);"  ##### "
250 PRINT TAB(X);"  # # # "
260 PRINT TAB(X);"  # # # "
270 PRINT TAB(X);"  #  "
280 PRINT TAB(X);"  # #  "
290 PRINT TAB(X);"  #  "
300 PRINT TAB(X);"  # #  "
310 PRINT TAB(X);"  V V  "
320 FOR D=1 TO 28
330 NEXT D
340 PRINT CHR$(19);
350 PRINT TAB(X);"  >  "
360 PRINT TAB(X);"  ##### "
370 PRINT TAB(X);"  # # # "
380 PRINT TAB(X);"  # # #"
390 PRINT TAB(X);"  #  "
400 PRINT TAB(X);"  #  "
410 PRINT TAB(X);"  #  "
420 PRINT TAB(X);"  #  "
430 PRINT TAB(X);"  V V  "
440 FOR D=1 TO 28
450 NEXT D
460 NEXT X
470 PRINT CHR$(19);
480 FOR X=1 TO 9
490 PRINT TAB(12);"  "
500 NEXT X
510 GOTO 40

```

ARC

RI

BY

The  
lot w  
just  
coun  
edg  
cap  
Rob  
pag  
have  
you  
Su  
of m  
hun  
firm  
cion  
is b  
you  
real  
this  
park  
lock  
De  
One  
be o  
door  
ter  
have  
find  
you  
door  
clea  
bot  
ever  
Yo  
gad  
joys  
keyt  
left  
right  
"M"  
(dov  
righ  
mak  
ing  
in th  
robo  
poin  
the  
mor  
posi  
to u  
trap  
He's  
look  
thin  
time  
PRO  
W  
gad  
ous  
mar  
mov  
it, n  
its p



# RENEGADE ROBOT II

BY JOEY LATIMER

The underground parking lot where you work seems just a little *too* quiet. Of course, you're still a bit edgy after your narrow escape from the *Renegade Robot* (FC, August 1984, page 74), but still . . . you have the creepy feeling that you're being watched.

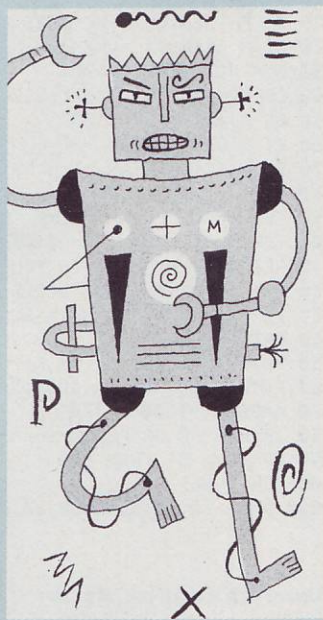
Suddenly, the hot smell of machine oil and the hum of metal wheels confirm your worst suspicions: The renegade robot is back, and it's time for you to *get lost!* But luck's *really* running against you this time—the doors of the parking lot all seem to be locked!

Don't give up hope. One of the doors may still be open. There are six doors around the perimeter of the lot, and you may have to try all of them to find out which one offers you an escape. Jiggling door handles and keeping clear of the renegade robot's steel pincers, however, isn't going to be easy.

You can escape in *Renegade Robot II* using your joystick (port 1) or your keyboard. Use the "U" (up left), "I" (up), "O" (up right), "J" (left), "L" (right), "M" (down left), comma (down), and period (down right) keys to move. To make things more interesting, the longer you remain in the lot (and out of the robot's clutches), the more points you'll gain. Since the robot always heads more or less towards your position, you may be able to use the parked cars to trap him—but watch out! He's smarter than he looks, and may do something unpredictable at any time!

## PROGRAM NOTES

What makes the renegade robot such a dangerous opponent, capable of marking your position and moving inexorably toward it, negotiating obstacles in its path? The answer is in



a simple routine that controls the way the robot "thinks."

To make the robot track you, the computer begins by subtracting each coordinate of the robot's position (RX and RY) from the corresponding coordinate of your position (PX or PY). The values that result may each be positive, negative, or zero, depending on the robot's position relative to your own; thus the arithmetic signs of these values (determined by BASIC's SGN function) each may be 1, 0, or -1. Adding the signs to the coordinates of the robot's current position results in a proposed new position (GX, GY) that will place the robot one step closer to you.

The next step the computer takes is to determine if the robot can move to that new position, or is prevented by some obstacle from doing so. The contents of the screen at the new position are looked at (by a PEEK, LOCATE, SCREEN, or other statement) and analyzed. If the proposed movement will not land the robot on an obstacle, the program continues.

If, on the other hand, an obstacle is encountered, the computer discards the new position and attempts to find a way around the

obstacle. It begins by randomly generating a direction, D, which may be -1 or +1. It then generates a new proposed position (GX, GY) by adding this value to either the robot's horizontal coordinate, RX, or its vertical coordinate, RY, depending on the value of a flag, F. (A flag is a variable that is given one of two possible values, depending on conditions. Flags are often used when a program must alternate between two possible courses of action, trying first one, then the other.) The value of flag F is changed so that if the proposed position to left or right is found to be blocked, the next position to be tried will be above or below, and vice versa.

By this semi-random process of "controlled blundering," coupled with an inex-

orable means of tracking your position, the robot will eventually find its way around almost any small obstacle.

## CUSTOMIZING THE PROGRAM

Increasing or decreasing the number of obstacles in the robot's path makes *Renegade Robot II* easier or more difficult to play. The variable RV in line 10 controls the random process that places obstacles on the screen. You may adjust gameplay as you like by changing RV to any decimal number between 0 (most obstacles) and 1 (no obstacles).

If you're an ambitious programmer, you might also try to change how frequently the robot moves. We'll leave that one for you to figure out.

## ADAM/Renegade Robot II

```

10 DIM dr(6,2):rv = 0.5
20 FOR i = 1 TO 9:bl$ = bl$+CHR$(32):NEXT i
30 FOR i = 1 TO 6:READ dr(i,1),dr(i,2):NEXT i
40 HOME:PRINT TAB(8);"RENEGADE ROBOT II"
50 VTAB 5:PRINT "PRESS LEFT TRIGGER TO BEGIN."
60 IF PDL(7) <> 1 THEN c = RND(1):GOTO 60
70 HOME:GR:COLOR= 9
80 HLIN 0,39 AT 0:HLIN 0,39 AT 1
90 HLIN 0,39 AT 38:HLIN 0,39 AT 39
100 VLIN 2,37 AT 0:VLIN 2,37 AT 1
110 VLIN 2,37 AT 38:VLIN 2,37 AT 39:COLOR= 8
120 FOR i = 1 TO 6:PLOT dr(i,1),dr(i,2):NEXT i
130 FOR i = 3 TO 35 STEP 2:FOR j = 3 TO 36
140 IF RND(1) > rv THEN COLOR= INT(RND(1)*7)+1:PLOT j,
i
150 NEXT j,i
160 VTAB 21:HTAB 11:PRINT "SCORE:"
170 px = INT(RND(1)*10)+2:py = (INT(RND(1)*18)+1)*2
180 rx = INT(RND(1)*10)+26:ry = (INT(RND(1)*18)+1)*2
190 sc = 0:do = INT(RND(1)*6)+1:ts = RND(1)*500+500
200 COLOR= 15:PLOT px,py
210 COLOR= 11:PLOT rx,ry
220 j = PDL(5)
230 nx = px+(j = 2 OR j = 3 OR j = 6)-(j > 6)
240 ny = py+(j = 4 OR j = 6 OR j = 12)-(j = 1 OR j = 3
OR j = 9)
250 IF px = nx AND py = ny THEN 380
260 IF nx < 1 OR nx > 38 OR ny < 1 OR ny > 38 THEN 490
270 p = SCRIN(nx,ny):IF p = 0 THEN 320
280 IF p <> 8 THEN 380
290 VTAB 22:HTAB 1
300 IF nx <> dr(do,1) OR ny <> dr(do,2) THEN PRINT "LO
CKED!":GOTO 360
310 PRINT "UNLOCKED!"
320 COLOR= 8*(px = dr(do,1) AND py = dr(do,2))
330 PLOT px,py
340 COLOR= 15:PLOT nx,ny:px = nx:py = ny
350 IF p <> 8 THEN 380
360 FOR d = 1 TO 200:NEXT d
370 VTAB 22:HTAB 1:PRINT SPC(9)
380 IF sc < ts THEN 480
390 qx = rx+SGN(px-rx):qy = ry+SGN(py-ry):f = 1
400 p = SCRIN(qx,qy):IF p = 0 OR p = 15 THEN 450

```



## ARCADE GAME

```

410 d = 2*INT(RND(1)*2)-1
420 IF f THEN qx = rx+d:qy = ry:GOTO 440
430 qy = ry+d:qx = rx
440 f = NOT f:GOTO 400
450 COLOR= 0:PLOT rx,ry:COLOR= 11:PLOT qx,qy
460 IF p = 15 THEN 520
470 rx = qx:ry = qy
480 sc = sc+10:HTAB 17:VTAB 21:PRINT sc:GOTO 220
490 TEXT:HOME:FOR i = 1 TO 7
500 PRINT CHR$(7);:FOR d = 1 TO 50:NEXT d,i
510 PRINT "YOU ESCAPED!":GOTO 540
520 TEXT:HOME:PRINT CHR$(7);
530 PRINT "SORRY, YOU WERE CAUGHT!"
540 PRINT:PRINT "YOUR SCORE WAS ";sc
550 IF sc > hs THEN hs = sc
560 PRINT:PRINT "THE HIGH SCORE IS ";hs
570 PRINT:PRINT "PRESS LEFT TRIGGER TO PLAY AGAIN"
580 PRINT "OR RIGHT TRIGGER TO QUIT."
590 IF PDL(7) = 1 THEN 70
600 IF PDL(9) <> 1 THEN 590
610 END
1000 DATA 20,1,38,12,38,26,20,38,1,12,1,26

```

### Apple/Renegade Robot II

```

10 DIM DR(6,2):RV = 0.5
20 FOR I = 1 TO 6:READ DR(I,1),DR(I,2):NEXT I
30 HOME:PRINT TAB(11);"RENEGADE ROBOT II"
40 VTAB 5:PRINT TAB(9);"PRESS ANY KEY TO BEGIN."
50 POKE -16368,0
60 IF PEEK(-16384) < 128 THEN Q = RND(1):GOTO 60
70 POKE -16368,0
80 PRINT:PRINT TAB(8);"DO YOU WANT TO USE THE"
90 PRINT TAB(4);"<K>EYBOARD OR THE <J>OYSTICK?";
100 GET K$:IF K$ <> "K" AND K$ <> "J" THEN 100
110 KF = (K$ = "K")
120 HOME:GR:COLOR= 9
130 HLIN 0,39 AT 0:HLIN 0,39 AT 1
140 HLIN 0,39 AT 38:HLIN 0,39 AT 39
150 VLIN 2,37 AT 0:VLIN 2,37 AT 1
160 VLIN 2,37 AT 38:VLIN 2,37 AT 39
170 COLOR= 8
180 FOR I = 1 TO 6:PLOT DR(I,1),DR(I,2):NEXT I
190 FOR I = 3 TO 35 STEP 2:FOR J = 3 TO 36
200 IF RND(1) > RV THEN COLOR= INT(RND(1)*7)+1:PLOT J,
I
210 NEXT J,I
220 VTAB 21:HTAB 16:PRINT "SCORE:"
230 PX=INT(RND(1)*11)+2:PY=(INT(RND(1)*18)+1)*2
240 RX=INT(RND(1)*11)+27:RY=(INT(RND(1)*18)+1)*2
250 SC = 0:DO = INT(RND(1)*6)+1:TS = RND(1)*500+500
260 COLOR= 15:PLOT PX,PY
270 COLOR= 11:PLOT RX,RY
280 IF KF THEN 310
290 NY = PY+(PDL(1) > 192)-(PDL(1) < 64)
300 NX = PX+(PDL(0) > 192)-(PDL(0) < 64):GOTO 350
310 K = PEEK(-16384)-128
320 POKE -16368,0
330 NY = PY+(K = 44 OR K = 46 OR K = 77)-(K = 73 OR K
= 79 OR K = 85)
340 NX = PX+(K = 46 OR K = 76 OR K = 79)-(K = 74 OR K
= 77 OR K = 85)
350 IF PX = NX AND PY = NY THEN 470
360 IF NX < 1 OR NX > 38 OR NY < 1 OR NY > 38 THEN 580
370 P = SCRN(NX,NY):IF P = 0 THEN 420
380 IF P <> 8 THEN 470
390 VTAB 21:HTAB 1
400 IF NX <> DR(D0,1) OR NY <> DR(D0,2) THEN PRINT "LO
CKED!":GOTO 450
410 PRINT "UNLOCKED!"
420 COLOR= 8*(PX = DR(D0,1) AND PY = DR(D0,2))
430 PLOT PX,PY:COLOR= 15:PLOT NX,NY:PX = NX:PY = NY
440 IF P <> 8 THEN 470
450 FOR D = 1 TO 200:NEXT D
460 VTAB 21:HTAB 1:PRINT SPC(9)
470 IF SC < TS THEN 570

```

```

480 QX = RX+SGN(PX-RX):QY = RY+SGN(PY-RY):F = 1
490 P = SCRN(QX,QY):IF P = 0 OR P = 15 THEN 540
500 D = 2*INT(RND(1)*2)-1
510 IF F THEN QX = RX+D:QY = RY:GOTO 530
520 QY = RY+D:QX = RX
530 F = NOT F:GOTO 490
540 COLOR= 0:PLOT RX,RY:COLOR= 11:PLOT QX,QY
550 IF P = 15 THEN 600
560 RX = QX:RY = QY
570 SC = SC+10:VTAB 21:HTAB 23:PRINT SC:GOTO 280
580 FOR D = 1 TO 50:A = PEEK(-16336):NEXT D
590 TEXT:HOME:PRINT:PRINT "YOU ESCAPED!":GOTO 620
600 TEXT:HOME:PRINT CHR$(7)
610 PRINT:PRINT "SORRY, YOU WERE CAUGHT!"
620 PRINT:PRINT "YOUR SCORE WAS ";SC
630 IF SC > HS THEN HS = SC
640 PRINT:PRINT "THE HIGH SCORE IS ";HS:PRINT
650 PRINT "PRESS <P> OR FIRE BUTTON TO PLAY AGAIN"
660 PRINT "OR <Q> TO QUIT.":POKE -16368,0
670 K=PEEK(-16384)-128
680 IF K = 80 OR PEEK(-16286) > 127 THEN 120
690 IF K <> 81 THEN 670
700 HOME:END
1000 DATA 20,1,38,12,38,26,20,38,1,12,1,26

```

### Atari/Renegade Robot II

```

10 DIM DR(6,2),BLS(9),BR$(38):RV=0.5
20 OPEN #1,4,0,"K":OPEN #6,12,0,"S:"
30 BLS=" ":BLS(9)=BLS:BLS(2)=BLS
40 BR$=CHR$(160):BR$(38)=BR$:BR$(2)=BR$
50 FOR I=1 TO 6:READ A,B:DR(I,1)=A:DR(I,2)=B:NEXT I
60 PRINT CHR$(125):POSITION 11,0
70 PRINT "RENEGADE ROBOT II"
80 POSITION 3,5
90 PRINT "DO YOU WANT TO USE THE <K>EYBOARD"
100 POSITION 10,6:PRINT "OR THE <J>OYSTICK?";
110 GET #1,K:IF K<>ASC("K") AND K<>ASC("J") THEN 110
120 KF=(K=ASC("K"))
130 PRINT CHR$(125);:POKE 82,0
140 POKE 752,1:SETCOLOR 1,0,15:SETCOLOR 2,0,0
150 POSITION 1,0:PRINT BR$:POSITION 1,1:PRINT BR$
160 FOR I=2 TO 20
170 POSITION 1,I:PRINT CHR$(160)
180 POSITION 38,I:PRINT CHR$(160)
190 NEXT I
200 POSITION 1,21:PRINT BR$:POSITION 1,22:PRINT BR$
210 FOR I=1 TO 6:POSITION DR(I,1),DR(I,2)
220 PRINT CHR$(35):NEXT I
230 FOR I=3 TO 19 STEP 2:FOR J=3 TO 36
240 IF RND(0)>RV THEN POSITION J,I:PRINT CHR$(219)
250 NEXT J:NEXT I:POSITION 15,23:PRINT "SCORE: 0";
260 PX=INT(RND(0)*10)+2:PY=(INT(RND(0)*9)+1)*2
270 RX=INT(RND(0)*10)+28:RY=(INT(RND(0)*9)+1)*2
280 SC=0:DO=INT(RND(0)*6)+1:TS=RND(0)*500+500
290 POSITION PX,PY:PRINT CHR$(27);CHR$(27)
300 POSITION RX,RY:PRINT CHR$(16)
310 IF KF THEN 360
320 J=STICK(0)
330 NX=PX+(J>=5 AND J<=7)-(J>=9 AND J<=11)
340 NY=PY+(J=5 OR J=9 OR J=13)-(J=6 OR J=10 OR J=14)
350 GOTO 390
360 K=PEEK(764):POKE 764,255
370 NX=PX+(K=0 OR K=8 OR K=34)-(K=1 OR K=11 OR K=37)
380 NY=PY+(K=32 OR K=34 OR K=37)-(K=8 OR K=11 OR K=13)
390 IF PX=NX AND PY=NY THEN 520
400 IF NX=0 OR NX=39 OR NY=0 OR NY=22 THEN 650
410 LOCATE NX,NY,P:POSITION NX,NY:PUT #6,P
420 IF P=32 THEN 470
430 IF P<>35 THEN 520
440 POSITION 0,23
450 IF NX<>DR(D0,1) OR NY<>DR(D0,2) THEN PRINT "LOCKED
!":GOTO 510
460 PRINT "UNLOCKED!";
470 POSITION PX,PY
480 PRINT CHR$(32+3*(PX=DR(D0,1) AND PY=DR(D0,2)))

```



```

490 POSITION NX,NY:PRINT CHR$(27);CHR$(27):PX=NX:PY=NY
500 IF P<>35 THEN 520
510 FOR D=1 TO 100:NEXT D:POSITION 0,23:PRINT BL$;
520 IF SC<TS THEN 640
530 QX=RX+SGN(PX-RX):QY=RY+SGN(PY-RY):F=1
540 LOCATE QX,QY,P:POSITION QX,QY:PUT #6,P
550 IF P=27 OR P=32 THEN 600
560 D=2*INT(RND(0)*2)-1
570 IF F THEN QX=RX+D:QY=RY:GOTO 590
580 QY=RY+D:QX=RX
590 F=NOT F:GOTO 540
600 POSITION RX,RY:PRINT CHR$(32)
610 POSITION QX,QY:PRINT CHR$(16)
620 IF P=27 THEN 700
630 RX=QX:RY=QY
640 SC=SC+10:POSITION 22,23:PRINT SC;:GOTO 310
650 SOUND 0,121,10,10:SOUND 1,96,10,10
660 SOUND 2,81,10,10
670 FOR D=1 TO 200:NEXT D
680 FOR I=0 TO 2:SOUND I,0,0,0:NEXT I
690 PRINT CHR$(125):PRINT "YOU ESCAPED!":GOTO 730
700 SOUND 0,45,12,10:FOR D=1 TO 200:NEXT D
710 SOUND 0,0,0,0
720 PRINT CHR$(125):PRINT "SORRY, YOU WERE CAUGHT!"
730 PRINT "PRINT \"YOUR SCORE IS \";SC;\".\":PRINT
740 IF SC>HS THEN HS=SC
750 PRINT "THE HIGH SCORE IS \";HS;\".\":PRINT :PRINT
760 PRINT "PRESS <P> OR THE FIRE BUTTON"
770 PRINT "TO PLAY AGAIN OR <Q> TO QUIT."
780 POKE 764,255
790 IF PEEK(764)=10 OR STRIG(0)=0 THEN POKE 764,255:GO
TO 130
800 IF PEEK(764)<>47 THEN 790
810 POKE 82,2:POKE 764,255:PRINT CHR$(125);:END
1000 DATA 20,1,38,8,38,16,20,21,1,16,1,8

```

### Commodore 64/Renegade Robot II

```

10 DIM DR(6,2):S=1024:C=55296:SD=54272:RV=0.5
20 BL$=CHR$(32):FOR I=1 TO 8:BL$=BL$+CHR$(32):NEXT I
30 FOR I=SD TO SD+23:POKE I,0:NEXT I
40 POKE SD+24,15:POKE SD+5,128:POKE SD+6,64
50 POKE 649,1:POKE 53280,12:POKE 53281,1
60 FOR I=1 TO 6:READ DR(I,1),DR(I,2):NEXT I
70 PRINT CHR$(147);TAB(11);CHR$(30);"RENEGADE ROBOT II"
"
80 PRINT:PRINT TAB(8);"PRESS ANY KEY TO BEGIN."
90 GET K$:IF K$="" THEN Q=RND(1):GOTO 90
100 PRINT
110 PRINT TAB(8);CHR$(28);"DO YOU WANT TO USE THE"
120 PRINT TAB(5);"<K>EYBOARD OR THE <J>OYSTICK?"
130 GET K$:IF K$<>"K" AND K$<>"J" THEN 130
140 KF=(K$="K")
150 PRINT CHR$(147):FOR I=0 TO 39
160 TB=S+I:BB=TB+920:LS=S+I*40:RS=LS+39
170 POKE TB,160:POKE TB+SD,2
180 POKE TB+40,160:POKE TB+SD+40,2
190 POKE BB,160:POKE BB+SD,2
200 POKE BB+40,160:POKE BB+SD+40,2
210 IF I>1 AND I<23 THEN POKE LS,160:POKE LS+SD,2:POKE
RS,160:POKE RS+SD,2
220 NEXT I
230 FOR I=1 TO 6:P=DR(I,1)+DR(I,2)*40
240 POKE S+P,102:POKE C+P,0:NEXT I
250 FOR I=3 TO 21 STEP 2:FOR J=2 TO 37
260 IF RND(1)>RV THEN POKE S+J+I*40,160:POKE C+J+I*40,
INT(RND(1)*13)+2
270 NEXT J,I
280 POKE 214,23:PRINT
290 PRINT TAB(15);CHR$(18);CHR$(28);"SCORE: 0";
300 PX=INT(RND(1)*10)+1:PY=(INT(RND(1)*10)+1)*2
310 RX=INT(RND(1)*10)+29:RY=(INT(RND(1)*10)+1)*2
320 SC=0:DO=INT(RND(1)*6)+1:TS=RND(1)*500+500
330 POKE S+PX+PY*40,81:POKE C+PX+PY*40,0
340 POKE S+RX+RY*40,90:POKE C+RX+RY*40,0
350 IF KF THEN 400

```

```

360 J=15-(PEEK(56321) AND 15)
370 NX=PX-(J>=8 AND J<=10)+(J>=4 AND J<=6)
380 NY=PY-(J=2 OR J=6 OR J=10)+(J=1 OR J=5 OR J=9)
390 GOTO 420
400 GET K$:NX=PX-(K$="O" OR K$="L" OR K$=".")+(K$="U"
OR K$="J" OR K$="M")
410 NY=PY-(K$="M" OR K$="," OR K$=".")+(K$="U" OR K$="
I" OR K$="O")
420 IF PX=NX AND PY=NY THEN 560
430 IF NX<0 OR NX>39 OR NY<1 OR NY>23 THEN 690
440 P=PEEK(S+NX+NY*40)
450 IF P=32 THEN 510
460 IF P<>102 THEN 560
470 POKE 214,23:PRINT
480 LF=(NX<>DR(D0,1) OR NY<>DR(D0,2))
490 IF LF THEN PRINT CHR$(18);"LOCKED!";:GOTO 540
500 PRINT CHR$(18);"UNLOCKED!";
510 POKE S+PX+PY*40,32-70*(PX=DR(D0,1) AND PY=DR(D0,2)
)
520 POKE S+NX+NY*40,81:POKE C+NX+NY*40,0:PX=NX:PY=NY
530 IF P<>102 THEN 560
540 FOR D=1 TO 200:NEXT D
550 POKE 214,23:PRINT:PRINT CHR$(18);BL$;
560 IF SC<TS THEN 670
570 QX=RX+SGN(PX-RX):QY=RY+SGN(PY-RY):F=-1
580 P=PEEK(S+QX+QY*40):IF P=81 OR P=32 THEN 630
590 D=2*INT(RND(1)*2)-1
600 IF F THEN QX=RX+D:QY=RY:GOTO 620
610 QY=RY+D:QX=RX
620 F=NOT F:GOTO 580
630 POKE S+RX+RY*40,32
640 POKE S+QX+QY*40,90:POKE C+QX+QY*40,0
650 IF P=81 THEN 740
660 RX=QX:RY=QY
670 SC=SC+10:POKE 214,23:PRINT
680 PRINT TAB(21);CHR$(18);SC;:GOTO 350
690 POKE SD+4,33:FOR I=1 TO 5:POKE SD+1,50
700 FOR D=1 TO 50:NEXT D:POKE SD+1,20
710 FOR D=1 TO 50:NEXT D:NEXT I:POKE SD+4,0
720 PRINT CHR$(147):PRINT:PRINT "YOU ESCAPED!"
730 GOTO 790
740 POKE SD+4,33:FOR I=1 TO 20
750 POKE SD,INT(RND(1)*2)+135
760 POKE SD+1,INT(RND(1)*2)+17:NEXT I:POKE SD+4,0
770 PRINT CHR$(147)
780 PRINT CHR$(31);"SORRY, YOU WERE CAUGHT!"
790 PRINT:PRINT CHR$(154);"YOUR SCORE WAS";SC
800 IF SC>HS THEN HS=SC
810 PRINT:PRINT "THE HIGH SCORE IS";HS:PRINT
820 PRINT "PRESS <P> OR FIRE BUTTON TO PLAY AGAIN"
830 PRINT "OR <Q> TO QUIT."
840 GET K$
850 IF K$="P" OR (PEEK(56321) AND 16)=0 THEN 150
860 IF K$<>"Q" THEN 840
870 PRINT CHR$(147);:END
1000 DATA 20,1,39,8,39,16,20,23,0,8,0,16

```

### IBM PC w/Color Graphics Adapter & IBM PCjr/ Renegade Robot II

```

9 REM --MAKE SURE YOU'RE IN ALL-CAPS MODE--
10 DIM DR(6,2):RV=.5
20 WIDTH 40:KEY OFF:CLS:SCREEN 0,1:STRIG ON
30 FOR I=1 TO 6:READ DR(I,1),DR(I,2):NEXT I
40 CLS:COLOR 7:PRINT TAB(11);"RENEGADE ROBOT II"
50 LOCATE 5,8,0:PRINT "PRESS ANY KEY TO BEGIN."
60 IF INKEY$="" THEN Q=RND:GOTO 60
70 PRINT:PRINT TAB(8);"DO YOU WANT TO USE THE"
80 PRINT TAB(5);"<K>EYBOARD OR THE <J>OYSTICK?"
90 K$=INKEY$:IF K$<>"K" AND K$<>"J" THEN 90
100 KF=(K$="K")
110 CLS:COLOR 9:PRINT STRING$(80,219);
120 FOR I=1 TO 19
130 PRINT CHR$(219);SPC(38);CHR$(219);
140 NEXT I:PRINT STRING$(80,219);
150 COLOR 8:FOR I=1 TO 6
160 LOCATE DR(I,2),DR(I,1)

```



## ARCADE GAME

```

170 PRINT CHR$(178):NEXT I
180 FOR I=4 TO 20 STEP 2:FOR J=3 TO 38
190 IF RND>RV THEN COLOR INT(RND*7)+1:LOCATE I,J:PRINT
CHR$(148)
200 NEXT J,I
210 LOCATE 23,16:COLOR 7:PRINT"SCORE: 0";
220 PX=INT(RND*11)+2:PY=(INT(RND*10)+1)*2+1
230 RX=INT(RND*11)+29:RY=(INT(RND*10)+1)*2+1
240 SC=0:DO=INT(RND*6)+1:TS=RND*500+500
250 COLOR 15:LOCATE PY,PX:PRINT CHR$(2)
260 COLOR 11:LOCATE RY,RX:PRINT CHR$(15)
270 IF KF THEN 310
280 JO=STICK(0):J1=STICK(1)
290 NX=PX-(JO>72)+(JO<24)
300 NY=PY-(J1>72)+(J1<24):GOTO 340
310 K$=INKEY$
320 NX=PX+(K$="U" OR K$="J" OR K$="M")-(K$="O" OR K$="
L" OR K$=".")
330 NY=PY+(K$="U" OR K$="I" OR K$="O")-(K$="M" OR K$="
" OR K$=".")
340 IF PX=NX AND PY=NY THEN 460
350 IF NX<1 OR NX>40 OR NY<2 OR NY>22 THEN 570
360 P=SCREEN(NY,NX):IF P=32 THEN 410
370 IF P<>178 THEN 460
380 LOCATE 24,1:COLOR 7
390 IF NX<>DR(DO,1) OR NY<>DR(DO,2) THEN PRINT "LOCKED
!";:GOTO 450
400 PRINT "UNLOCKED!";
410 EH=(PX=DR(DO,1) AND PY=DR(DO,2)):COLOR -8*EH
420 LOCATE PY,PX:PRINT CHR$(32-146*EH)
430 COLOR 15:LOCATE NY,NX:PRINT CHR$(2):PX=NX:PY=NY
440 IF P<>178 THEN 460
450 FOR D=1 TO 200:NEXT D:LOCATE 24,1:PRINT SPC(9);
460 IF SC<TS THEN 560
470 QX=RX+SGN(PX-RX):QY=RY+SGN(PY-RY):F=-1
480 P=SCREEN(QY,QX):IF P=32 OR P=2 THEN 520
490 D=2*INT(RND*2)-1:IF F THEN QX=RX+D:QY=RY:GOTO 510
500 QY=RY+D:QX=RX
510 F=NOT F:GOTO 480
520 LOCATE RY,RX:PRINT CHR$(32)
530 COLOR 11:LOCATE QY,QX:PRINT CHR$(15)
540 IF P=2 THEN 600
550 RX=QX:RY=QY
560 SC=SC+10:LOCATE 23,22:COLOR 7:PRINT SC;:GOTO 270
570 FOR S=1 TO 5
580 SOUND 523,1:SOUND 659,1:SOUND 784,1:NEXT S
590 CLS:COLOR 7:PRINT:PRINT "YOU ESCAPED!":GOTO 630
600 FOR I=1 TO 5
610 FOR J=800 TO 400 STEP -10:SOUND J,.2:NEXT J,I
620 CLS:COLOR 7:PRINT:PRINT "SORRY, YOU WERE CAUGHT!"
630 PRINT:PRINT:PRINT "YOUR SCORE IS";SC
640 IF SC>HS THEN HS=SC
650 PRINT:PRINT "THE HIGH SCORE IS";HS
660 PRINT
670 PRINT "PRESS <P> OR FIRE BUTTON TO PLAY AGAIN"
680 PRINT "OR <Q> TO QUIT."
690 K$=INKEY$
700 IF K$="P" OR STRIG(0)=-1 THEN 110
710 IF K$<>"Q" THEN 690
720 CLS:COLOR 7:END
1000 DATA 20,2,40,8,40,16,20,22,1,16,1,8

```

### TRS-80 Color Computer/Renegade Robot II

```

10 DIM DR(6,2):S=1024:RV=0.4
20 FOR I=1 TO 9:BL$=BL$+CHR$(128):NEXT I
30 FOR I=1 TO 6:READ DR(I,1),DR(I,2):NEXT I
40 CLS:PRINT TAB(7);"RENEGADE ROBOT II"
50 PRINT068,"PRESS ANY KEY TO BEGIN."
60 IF INKEY$="" THEN Q=RND(1):GOTO 60
70 PRINT:PRINT:PRINT TAB(4);"DO YOU WANT TO USE THE"
80 PRINT " <K>EYBOARD OR THE <J>OYSTICK?"
90 K$=INKEY$:IF K$<"K" AND K$<"J" THEN 90
100 KF=(K$="K")
110 CLS
120 FOR I=0 TO 31:TB=S+I:BB=TB+448:LS=S+I*32:RS=LS+31

```

```

130 POKE TB,128
140 POKE BB,128:POKE BB+32,128
150 IF I>0 AND I<14 THEN POKE LS,128:POKE RS,128
160 NEXT I
170 FOR I=1 TO 6:P=DR(I,1)+DR(I,2)*32
180 POKE S+P,255:NEXT I
190 FOR I=2 TO 12 STEP 2:FOR J=2 TO 28
200 IF RND(10)>RV*10 THEN POKE S+J+I*32,143+RND(6)*16
210 NEXT J,I:PRINT @491,"SCORE: 0";
220 PX=RND(6):PY=RND(7)*2-1
230 RX=RND(6)+24:RY=RND(7)*2-1
240 SC=0:DO=RND(6):TS=RND(500)+500
250 POKE S+PX+PY*32,106
260 POKE S+RX+RY*32,79
270 IF KF THEN 310
280 JO=JOYSTK(0):J1=JOYSTK(1)
290 NX=PX+(JO<20)-(JO>43)
300 NY=PY+(J1<20)-(J1>43):GOTO 340
310 K$=INKEY$
320 NX=PX-(K$="O" OR K$="L" OR K$=".")+(K$="U" OR K$="
J" OR K$="M")
330 NY=PY-(K$="M" OR K$="," OR K$=".")+(K$="U" OR K$="
I" OR K$="O")
340 IF PX=NX AND PY=NY THEN 460
350 IF NX<0 OR NX>31 OR NY<0 OR NY>14 THEN 570
360 P=PEEK(S+NX+NY*32)
370 IF P=96 THEN 420
380 IF P<>255 THEN 460
390 PRINT @481,"";
400 IF NX<>DR(DO,1) OR NY<>DR(DO,2) THEN PRINT "LOCKED
!";:GOTO 450
410 PRINT "UNLOCKED!";
420 POKE S+PX+PY*32,96-159*(PX=DR(DO,1) AND PY=DR(DO,2
))
430 POKE S+NX+NY*32,106:PX=NX:PY=NY
440 IF P<>255 THEN 460
450 FOR D=1 TO 200:NEXT D:PRINT@481,BL$;
460 IF SC<TS THEN 560
470 QX=RX+SGN(PX-RX):QY=RY+SGN(PY-RY):F=-1
480 P=PEEK(S+QX+QY*32):IF P=106 OR P=96 THEN 530
490 D=2*INT(RND(0)*2)-1
500 IF F THEN QX=RX+D:QY=RY:GOTO 520
510 QY=RY+D:QX=RX
520 F=NOT F:GOTO 480
530 POKE S+RX+RY*32,96:POKE S+QX+QY*32,79
540 IF P=106 THEN 590
550 RX=QX:RY=QY
560 SC=SC+10:PRINT @497,SC;:GOTO 270
570 FOR I=1 TO 40:SOUND I,1:NEXT I
580 CLS:PRINT "YOU ESCAPED!":GOTO 610
590 SOUND 127,10
600 CLS:PRINT:PRINT "SORRY, YOU WERE CAUGHT!"
610 PRINT:PRINT "YOUR SCORE WAS";SC
620 IF SC>HS THEN HS=SC
630 PRINT:PRINT "THE HIGH SCORE IS";HS
640 PRINT:PRINT "PRESS <P> OR FIRE BUTTON TO"
650 PRINT "PLAY AGAIN OR <Q> TO QUIT."
660 K$=INKEY$:IF K$="P" OR PEEK(65280)=254 THEN 110
670 IF K$="Q" THEN END ELSE 660
1000 DATA 16,0,31,3,31,11,16,14,0,3,0,11

```

### TRS-80 Model III/Renegade Robot II

```

9 REM --THIS VERSION USES KEYBOARD ONLY--
10 DIM DR(6,2):S=15360:RV=0.5:PRINT CHR$(21);
20 FOR I=1 TO 6:READ DR(I,1),DR(I,2):NEXT I
30 CLS:PRINT TAB(25);"RENEGADE ROBOT II"
40 PRINT0342,"PRESS ANY KEY TO BEGIN."
50 IF INKEY$="" THEN Q=RND(0):GOTO 50
60 CLS:FOR I=0 TO 63
70 TB=S+I:BB=TB+896:LS=S+I*64:RS=LS+63
80 POKE TB,191:POKE TB+64,191
90 POKE BB,191:POKE BB+64,191
100 IF I>1 AND I<14 THEN POKE LS,191:POKE RS,191
110 NEXT I
120 FOR I=1 TO 6:P=DR(I,1)+DR(I,2)*64

```



```

130 POKE S+P,128:NEXT I
140 FOR I=3 TO 13 STEP 2:FOR J=2 TO 59
150 IF RND(10)>RV*10 THEN POKE S+J+I*64,7
160 NEXT J,I:PRINT @987,"SCORE: 0";
170 PRINT@864," ";
180 PX=RND(10)+1:PY=RND(6)*2
190 RX=RND(10)+52:RY=RND(6)*2
200 SC=0:DO=RND(6):TS=RND(500)+500
210 POKE S+PX+PY*64,253
220 POKE S+RX+RY*64,254
230 K$=INKEY$
240 NX=PX-(K$="O" OR K$="L" OR K$=".")+(K$="U" OR K$="
J" OR K$="M")
250 NY=PY-(K$="M" OR K$="," OR K$=".")+(K$="U" OR K$="
I" OR K$="O")
260 IF PX=NX AND PY=NY THEN 380
270 IF NX<0 OR NX>63 OR NY<1 OR NY>15 THEN 490
280 P=PEEK(S+NX+NY*64)
290 IF P=32 THEN 340
300 IF P<>128 THEN 380
310 PRINT @962,"";
320 IF NX<>DR(D0,1) OR NY<>DR(D0,2) THEN PRINT "LOCKED
!";GOTO 370
330 PRINT "UNLOCKED!";
340 POKE S+PX+PY*64,32-96*(PX=DR(D0,1) AND PY=DR(D0,2)
)
350 POKE S+NX+NY*64,253:PX=NX:PY=NY
360 IF P<>128 THEN 380
370 FOR D=1 TO 200:NEXT D:PRINT @962,STRING$(9,191);
380 IF SC<TS THEN 480
390 QX=RX+SGN(PX-RX):QY=RY+SGN(PY-RY):F=-1
400 P=PEEK(S+QX+QY*64):IF P=253 OR P=32 THEN 450
410 D=2*INT(RND(0)*2)-1
420 IF F THEN QX=RX+D:QY=RY:GOTO 440
430 QY=RY+D:QX=RX
440 F=NOT F:GOTO 400
450 POKE S+RX+RY*64,32:POKE S+QX+QY*64,254
460 IF P=253 THEN 500
470 RX=QX:RY=QY
480 SC=SC+10:PRINT @993,SC;:GOTO 230
490 CLS:PRINT:PRINT "YOU ESCAPED!":GOTO 510
500 CLS:PRINT:PRINT "SORRY, YOU WERE CAUGHT!"
510 PRINT:PRINT "YOUR SCORE WAS";SC
520 IF SC>HS THEN HS=SC
530 PRINT:PRINT "THE HIGH SCORE IS";HS
540 PRINT:PRINT "PRESS <> TO PLAY AGAIN"
550 PRINT "OR <Q> TO QUIT."
560 K$=INKEY$:IF K$="P" THEN 60
570 IF K$="Q" THEN END ELSE 560
1000 DATA 32,1,63,4,63,12,32,14,0,4,0,12

```

## PROGRAMMING P.S.

Correction to a previous program

**Atari/Banner** (April 1985, page 64)

As published, this program always uses asterisks when it draws the letters in your banner. To make it use the character of your choice, change line 630 to read as follows:

```
630 IF V=>BIN(K) THEN V=V-BIN(K):C$=CH$:GOTO 650
```

Correction to a program from "Helpful Hints"

**Kaypro or Other CP/M Machine/Translate BASIC Programs** (April 1985, page 51)

A parenthesis was omitted from the Kaypro equivalent of the Model III statement PRINT@W.... The correct Kaypro substitution for this statement is

```
PRINT CHR$(30);STRING$(W-INT(W/64)*64,12);STRING$(IN
T(W/64),10);...
```

# HOME INFORMATION MANAGER For the C 64

**PROGRAM BY STEVEN C.M. CHEN**  
**INTRODUCTION BY LANCE PAAVOLA**

Does your address book have so many crossouts and erasures that you can hardly read it? Can you tell how many times you've made a casserole by how much of the recipe card is obscured by splotches and spills?

With *Home Information Manager*, you can gather your files onto a slim, neat, portable disk or cassette, yet print out a "hard copy" whenever you want.

Electronic filing programs like this one are very versatile; their use is limited only by your imagination. Some of the most sophisticated file-management programs, called data-base management systems, pack enough power to handle all the computing needs of an entire business. (See *Working at Home*, page 12.)

But if you're not ready yet to buy a commercial program—maybe you're unsure whether storing your records electronically makes sense, or you think all you'll want to computerize is your 100-name Christmas card list—try *Home Information Manager*.

### THE FIRST STEPS

Begin by carefully typing in and *SAVING* the program. (See *Tips to the Typist*, page 53, for help with typing in programs.) Before you *RUN* the program, make sure your printer (if you have one) and disk drive or Datas-

ette are connected and turned on.

Don't be discouraged if it doesn't *RUN* the first time; with such a long program, you're bound to make typing errors. *LIST* your program to printer and proof-read it carefully. When you've got the program running, *SAVE* it and make a backup.

You can save your own data on the same disk or cassette, but you might want to start right off keeping data on separate disks or tapes. If that's your choice, get a blank tape or disk now, and format the disk. The name of the disk will appear on the screen when you list your fileboxes, so if you're going to have several disks for your data, name each appropriately (e.g., MOM, DAD or FINANCIAL, PERSONAL).

Before you transfer important records to your C 64, try out the program by creating a few sample fileboxes and filling in some info. When you're confident that you've located any remaining typing errors and have a good idea of how the program operates, you're ready to start using it in earnest.

### A COMPUTERIZED FILEBOX

To make *Home Information Manager* easy to learn, we've designed it to work just like a recipe box filled with index cards. You can have as many "fileboxes" as will fit on your